



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## **MODEL LABYRINTU**

MODEL OF LABYRINTH

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**PAVEL STEININGER**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. RADEK ŠTOHL, Ph.D.**

BRNO 2013



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
**Automatizační a měřicí technika**

**Student:** Pavel Steininger

**Ročník:** 3

**ID:** 115279

**Akademický rok:** 2012/2013

**NÁZEV TÉMATU:**

**Model labyrintu**

## POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s PLC firmy Rockwell Automation a vývojovým prostředím RS Logix 5000.
2. Seznamte se s modelem manipulátoru.
3. Realizujte program pro průchod labyrintem pomocí manipulátoru.
4. Vytvořte vizualizaci celého systému.
5. Ověřte funkčnost.

## DOPORUČENÁ LITERATURA:

Logix5000 Controllers General Instructions (Reference Manual). Milwaukee: Rockwell Automation, Inc. 2008.

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 27.5.2013

**Vedoucí práce:** Ing. Radek Štohl, Ph.D.

**Konzultanti bakalářské práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Bakalářská práce pojednává o naprogramování modelu labyrintu a komunikaci pomocí sítě AS-Interface, programu řízení modelu manipulátoru, kterým je simulován pohyb figurky v labyrintu.

Seznamuje s jednotlivými komponenty obou modelů a ukazuje postup pro konfiguraci programovacího prostředí RS Logix 5000 firmy Allen-Bradley.

Na závěr předvádí jednu z možností vizualizace systému.

## **Klíčová slova**

PLC, řízení, model labyrintu, model manipulátoru, vizualizace, Allen-Bradley, ControlLogix, FactoryTalk View, AS-Interface.

## **Abstract**

The bachelor's thesis deals with the programming of labyrinth model and communication using the AS-Interface network, the control program of manipulator model, which is simulated the movement of figure in the labyrinth.

It introduces the individual components of both models and shows how to configure the program RS Logix 5000 of Allen-Bradley company.

Finally demonstrates one of the visualization system options.

## **Keywords**

PLC, kontrol, model labyrinth, model manipulátor, visualization, Allen-Bradley, ControlLogix, FactoryTalk View, AS-Interface

### **Bibliografická citace:**

STEININGER, P. *Model labyrinthu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 50 s. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D.

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma Model labyrintu jsem vypracoval samostatně pod vedením vedoucího bakalářské práce, s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **24. května 2013**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing. Radku Štohlvi, Ph.D., za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

Také bych rád poděkoval rodičům za podporu při vytváření této práce a hlavně své přítelkyni za odbornou pomoc s formulací a pravopisem textu.

V Brně dne: **24. května 2013**

.....  
podpis autora

# Obsah

1	ÚVOD.....	9
2	TEORETICKÝ ÚVOD.....	10
2.1	PROGRAMOVATELNÝ AUTOMAT .....	10
2.1.1	Rozdělení PLC.....	10
2.1.2	Struktura PLC.....	11
2.2	AS-INTERFACE.....	12
2.2.1	Vlastnosti a parametry AS-I .....	12
2.2.2	Systém komunikace .....	13
2.2.3	Kabel AS-Interface .....	14
2.2.4	Zdroj napájení AS-Interface .....	14
2.3	SOUŘADNÝ SYSTÉM.....	15
3	PŘÍSTROJOVÉ VYBAVENÍ.....	16
3.1	PROGRAMOVATELNÝ AUTOMAT .....	16
3.1.1	Šasi 1756-A7 se zdrojem.....	16
3.1.2	Procesor 1756 – L62S .....	16
3.1.3	Modul EtherNet/IP 1756-ENBT/A .....	17
3.1.4	Modul DeviceNet 1756-DNB.....	17
3.1.5	Modul AS-i 3.0 Master/Scanner .....	17
3.1.6	Procesor 1756-L73.....	18
3.1.7	Modul SERCOS interface .....	18
3.2	ŘÍZENÍ .....	19
3.2.1	Kinetix 2000.....	19
3.2.2	Servomotory řady MPL.....	19
3.3	MODEL LABYRINTU .....	20
3.4	Konstrukce .....	20
3.5	Komponenty .....	21
3.5.1	Slave AC 5213.....	22
3.5.2	Slave AC5210.....	22
3.5.3	AC2088 a AC5000.....	23
3.5.4	Vyhodnocovací systém OV5012 .....	23
3.5.5	Snímač OE0004.....	24
3.5.6	Snímač OE0003 a OE0002.....	24

3.5.7	Napájení .....	25
3.6	MODEL MANIPULÁTORU .....	26
3.7	I/O modul 1971DS-IB8XOB8.....	26
4	PROGRAMOVÁNÍ.....	27
4.1	PROSTŘEDÍ PROGRAMU RS LOGIX 5000 .....	27
4.1.1	Založení projektu .....	27
4.1.2	Propojení s PLC .....	28
4.1.3	Struktura projektu .....	28
4.1.4	Online panel.....	28
4.1.5	Okno hlavního programu.....	29
4.1.6	Panel nástrojů prvků zvoleného programovacího jazyka .....	29
4.1.7	Informační okno.....	29
4.1.8	Definice proměnných (tagů) .....	30
4.1.9	Programovací jazyk .....	30
4.2	PROGRAM MODELU LABYRINTU.....	31
4.2.1	Přidání modulu AS-I .....	32
4.2.2	Přístup ke slovům.....	32
4.2.3	Použité funkce .....	32
4.2.4	Vlastní program labyrintu .....	33
4.3	PROGRAM MODELU MANIPULÁTORU .....	36
4.3.1	Konfigurace .....	36
4.3.2	Použité funkce .....	38
4.3.3	Vlastní program manipulátoru.....	41
4.4	VIZUALIZACE .....	44
4.4.1	Konfigurace .....	44
4.4.2	Vlastní vizualizace .....	45
5	Závěr .....	47
6	Literatura.....	48
7	Seznam příloh.....	50



# 1 ÚVOD

Úkolem této práce je seznámit se s PLC firmy Rockwell Automation a jejím vývojovým prostředím pro programování logických automatů RS Logix5000. Dále se seznámit s modelem manipulátoru a jeho instrumentací, tak aby bylo možné realizovat pomocí tohoto manipulátoru průchod modelem labyrintu. Posledním úkolem této práce je vytvořit vizualizaci celého systému, díky které bude umožněno ovládání.

Tato práce se tedy zprvu zabývá uvedením jednotlivých komponent systému a problematikou zajištění funkčnosti. Uvádí instrumentaci celého systému a komunikaci jednotlivých komponent. Představuje komunikační síť AS-interface jako jedno z možných řešení komunikace s periferiemi automatu. Dále se zabývá problematikou programování logických automatů a vzájemnou komunikací mezi nimi, tak aby pokyny jednoho mohl být řízen druhý.

Práce řeší naprogramování automatu pro navigaci pohybu manipulátoru v předem daném modelu labyrintu na základě zadaného algoritmu správného projití labyrintem, naprogramování automatu pro řízení tříosého modelu manipulátoru, který slouží k pohybu figurkou modelem labyrintu.

Nakonec je v práci řešeno zajištění vizualizace pro celý systém, a také potřebné ovládání pomocí vizualizace.

## 2 TEORETICKÝ ÚVOD

### 2.1 PROGRAMOVATELNÝ AUTOMAT [1]

Programovatelný logický automat neboli PLC (z anglického překladu Programmable Logic Controller) je zařízení, které zajišťuje sběr signálů přicházejících z technologického procesu na jeho vstupy, a na základě sekvenčních logických a časových funkcí posílá na své výstupy povely (výstupní signály), kterými se ovládají motory, klapky, ventily, topná tělesa apod.

PLC jsou nejvýznamnějším řídicím prostředkem pro řízení technologických procesů, výrobních linek a strojů.

#### 2.1.1 Rozdělení PLC

Výrobci programovatelných automatů dodávají velkou škálu programovatelných automatů, které se liší konstrukčně i svými parametry, z nichž nejdůležitější jsou:

- Velikost / výkon
- Integrované rozhraní
- Integrovaný LCD displej
- Modularita
- Integrované hodiny reálného času

##### Velikost a výkon

Podle velikosti se PLC dělí na malé, střední a velké. Velikost není dána jen konstrukčními rozměry, ale i výkonem, který opět není určen jedním parametrem, nýbrž celou řadou parametrů jako jsou:

- Velikost uživatelské a bitové paměti
- Počet připojitelných vstupů a výstupů
- Rychlost zpracování instrukcí
- Počet čítačů, časovačů

##### Modularita

**Kompaktní PLC** – setkáváme se s nimi jen u malých PLC. Mají v jednom tělese integrovaný centrální procesor i určitý počet vstupů a výstupů. Většinou je možné k nim připojit další moduly vstupů a výstupů. Kompaktní programovací automaty mohou ale sloužit i jako komponenty v distribuovaných řídicích systémech.

**Modulární PLC** - jsou vhodné pro automatizační úlohy středního a velkého rozsahu. Jsou tvořeny v podstatě pevným procesorovým jádrem s napájecím zdrojem umístěným v rámu, ke kterému se přes sběrnici připojují místní i vzdálené periferní jednotky. Nezbytným doplňkem modulárního programovatelného automatu jsou také ovládací panely, datové terminály a vizualizační prostředky.

### Integrované rozhraní

Všechny programovatelné automaty mají základní rozhraní, kterým komunikují s programovacím počítačem. Jedná se především o:

- Multi-Point Interface (MPI)
- PROFIBUS (Fieldbus)
- PROFINET (IndustrialEthernet)

### Integrované hodiny reálného času

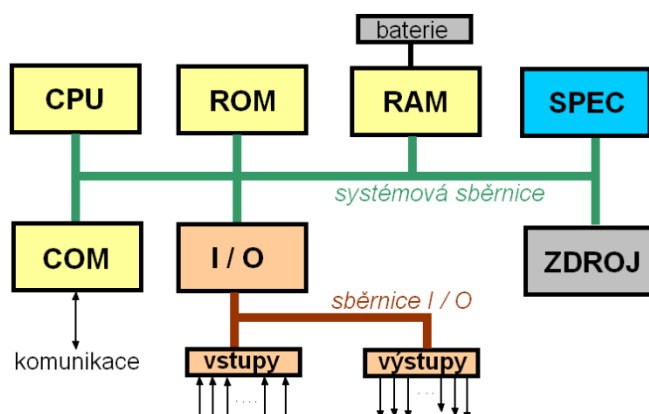
Malé PLC většinou hodiny reálného času nemají, střední obvykle ano, u velkých jsou hodiny samozřejmostí.

### Integrovaný LCD displej

Poznatky z aplikací PLC ukázaly velký rozsah aplikací výkonově nižších PLC střední velikosti doplněných jednoduchým operátorským panelem s textovým displejem. Proto se začal na PLC dávat integrovaný LCD textový nebo i grafický displej.

## 2.1.2 Struktura PLC [4]

Bloková struktura programovatelného automatu je znázorněna na obr. 1.1. Základem PLC jsou v principu tři funkční bloky: zpracování informace, vstupy/výstupy a paměť. Ostatní bloky jsou podpůrné a doplňkové.



Obrázek 2.1bloková struktura PLC [4]

Jednotlivé bloky uvedené na obrázku mají následující funkci:

**I/O** - jednotka řízení vstupů a výstupů snímá hodnoty vstupních veličin z procesu, konvertuje je do číslkové formy a ukládá do operační paměti,

**CPU**-centrální jednotka zpracovává informace, tj. podle

programu čte hodnoty vstupních a pomocných proměnných, provádí s nimi logické operace,

**RAM** - operační paměť slouží k ukládání řídicího programu a hodnot vstupních, výstupních a pomocných proměnných,

**ROM** - paměť pouze pro čtení obsahuje systémové programy, které uživatel nemůže modifikovat,

**SPEC** - speciální funkce obsahuje modul hodin, časovače, čítače, sekvenční registry, matematické funkce apod.,

**COM** - zajišťuje komunikaci automatu s okolím po sériové lince nebo po speciální komunikační sběrnici.

## 2.2 AS-INTERFACE [2]

Sběrnice AS-Interface (Actuator Sensor Interface) je inteligentní komunikační systém pro nejnižší úroveň automatizace. Byla původně navržena pro přenos převážně binárních signálů s cílem propojit mnohobodovým digitálním spojem především instrumentaci systémů logického řízení.

Síť AS-interface je typu master-slave (multislave), k síti je připojeno jedno zařízení typu master a až 62 zařízení typu slave. V původní verzi byla možnost připojit 31 zařízení typu slave, ale nová specifikace AS-Interface verze 2.1 rozšířila počet zařízení slave. Slave umožňuje připojení až 4 binárních signálů, tedy prakticky jedno zařízení typu master může komunikovat se 124, resp. 248 pasivními binárními vstupními a výstupními body. V roce 2004 byla vytvořena nová specifikace standardu AS-I verze 3.0 [3], která rozšiřuje možnosti sběrnice o přenos 16bitových informací ve více cyklech i jednom cyklu za cenu snížení maximálního počtu slávů v síti.

AS-I slave smí ze sítě AS-I odebírat max. 200 mA (ve verzi 2.1 je to pouze 100mA). Vyžaduje-li větší napájecí proudy, je možné použít přídatný napájecí rozvod 24V ss nebo rozvod 110V/230V.

Master síť AS-I je obvykle realizován jako modul do PLC nebo stand – alone master nebo jako gateway (mezisběrnicový spoj) s integrovaným malým PLC.

Kabel, který se stal obchodní značkou AS-Interface, je žluté barvy a přenáší společně napájení i data. Další jeho výhodou je snadné připojení všech komponentů pomocí využití prořezávací metody, díky které nepřehledná kabelová změť a objemné rozvaděče patří minulosti.

### 2.2.1 Vlastnosti a parametry AS-I

**Kabelová sběrnice** - Kabelovou sběrnici AS-Interface tvoří nestíněný nekroucený dvoužilový kabel, který zajišťuje přenos datových zpráv i napájecího napětí. Kromě přenosu dat slouží kabel také k napájení slávů elektrickou energií (obvykle 24 V DC). Jeho izolace je zvláštním způsobem profilovaná tak, aby umožnila snadnou, spolehlivou a rychlou montáž. Je-li to zapotřebí, používá se pomocný napájecí kabel, který má podobné vlastnosti.

**Uspořádání** - V síti je jen jedna řídicí stanice – Master. Master, postupně podle jejich přiřazené adresy, vyzývá všechny ostatní účastníky (podřízené stanice) – Slavy.

**Adresace** - Každá podřízená stanice (slave) má přiřazenou trvalou adresu. Adresy může přiřazovat master nebo přiřazovací přístroj, z výroby je nastavena adresa 0. Adresa je pevná a jednoznačná a je zapsána v každém modulu v energeticky nezávislé paměti EEPROM.

**Délka sítě (vedení)** - Délka kabelu může být až 100 m. Tato velikost je fyzikální omezení, a je kompromisem, který byl přijat během vývoje. Kompromis, který byl přijat za cenu struktury kabelu, cílové doby cyklu, možnosti výběru topologie a počtem účastníků. V případě potřeby lze použít repeatery (opakovače), které umožňují prodloužit vedení skoro až na 500 m.

**Počet I/O** - Ke každé aktivní řízené stanici lze připojit až čtyři binární vstupní nebo výstupní pasivní prvky (spínače, binární výstupy), nebo jejich kombinace. U všech modulů typu slave to tedy činí podle specifikace v1.0 až 124 účastníků na jeden sběrnice svazek, podle v2.1 to činí 248 vstupů a 186 výstupů. Podle v3.0 248 vstupů a 248 výstupů.

**Bezpečnost** - AS-Interface nemusí obsahovat ochranný vodič PE, čímž se ušetří na nákladech. Podle specifikací je každá síť konfigurována jako soustava PELV podle normy IEC 60364-4-41 (Protective Extra Low Voltage),[6].

## 2.2.2 Systém komunikace

Stejně jako všechny průmyslové sběrnice lze i AS-Interface zařazovat podle sedmi úrovní referenčního modelu ISO/OSI [5]. AS-Interface využívá z těchto sedmi úrovní pouze tři:

**1:Fyzická vrstva** - je orientována na elektrické a mechanické spojení účastníků komunikace, a přenáší tok informace mezi nimi.

**2:Linková vrstva** - je nad fyzickou vrstvou a odpovídá za spolehlivý přenos dat. V této vrstvě jsou data formátována do rámců, opatřena informací o adrese a informací o zabezpečení dat, a postupně vysílána po vedení.

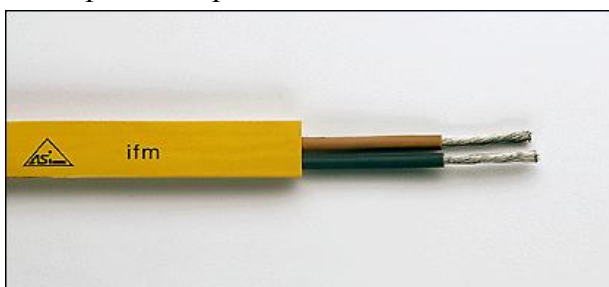
**7:Aplikační vrstva**- definuje povely, obsahy dat, posloupnost cyklu AS-I a chování účastníků, například při výměně a připojení slavnů za provozu systému.

Zbývající čtyři vrstvy ISO/OSI modelu jsou vynechány.

### 2.2.3 Kabel AS-Interface[7]

Jako přenosové médium byly pro AS-Interface definovány dva různé typy dvojvodičových kabelů:

- Nízkokapacitní kabel (vyhovující normám CENELEC a DIN VDE0281), jde o levný běžně dodávaný typ kabelu.
- Speciální plochý kabel, který má podobné elektrické vlastnosti, jako předešlý typ, ale má speciální přednosti pro montáž a instalaci



Obrázek 2.2 Sběrníkový kabel AS-Interface se speciálním profilem [7]

Sběrníkový kabel pro AS-Interface je známý jako nestíněný dvou vodičový kabel, který jak již bylo zmíněno, přenáší data i napájení. Jeho vnější obal je žluté barvy a má speciálně tvarovaný plášť, díky kterému nemůže dojít k záměně polarity. Průřez vodičů je 1,5 mm<sup>2</sup>, jsou svíjené z měděných jemných drátků, třídy 6 podle DIN VDE0295, tloušťka stěny izolace je 0,5 mm. Vodič s kladným napětím má hnědou izolaci, se záporným napětím světle modrou.

Někdy nepostačuje tento způsob napájení ze sítě AS-Interface, nebo může být jiný důvod pro oddělení přenosu dat a napájení. K tomu se používá pomocný napájecí kabel černé barvy, který se zapojuje stejně jednoduchým způsobem.

### 2.2.4 Zdroj napájení AS-Interface

Zdroj napájení musí splňovat 4 úlohy:

- Slouží pro napájení slavů a části masteru po dvou vodičovém kabelu. Zdroj poskytuje stejnosměrné napětí v rozsahu 29,5 V až 31,6 V při proudovém zatížení max 8 A.
- Slouží pro symetrické vyvážení sítě AS-I, protože celá síť pracuje jako symetrická neuzemněná soustava. Symetrizaci zajišťuje symetrizační obvod zdroje napájení.

- Dále slouží pro oddělení dat. Tento obvod pro oddělení dat je obvykle umístěn v krytu společně se zdrojem napětí, a skládá se ze dvou tlumivek s indukčností 50mH a dvou paralelně zapojených rezistorů s odporem  $39\Omega$ .
- Poslední úloha zdroje má bezpečnostní význam. Systém AS-I je navržen pro malá napětí s bezpečným oddělením (PELV,[6]). To podle norem IEC znamená, že se vyžaduje, aby síť AS-I byla bezpečně oddělena od napájecí sítě.

## 2.3 SOUŘADNÝ SYSTÉM

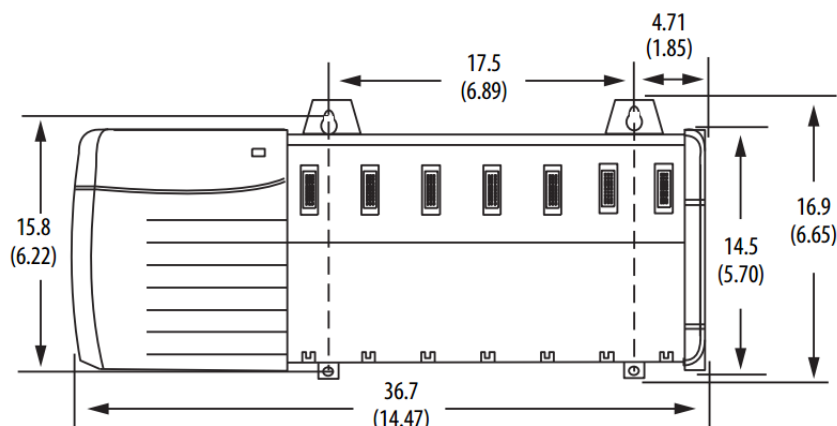
Slouží k určování souřadnic polohy tělesa či bodu ve zvolené soustavě. Hlavním parametrem souřadného systému je počet os neboli stupeň volnosti. Tedy počet os a jejich vzájemný vztah. Model manipulátoru je takzvaný jeřábový manipulátor, který obsahuje tři translační osy X, Y a Z. Díky tomuto spojení dostáváme kartézskou soustavu souřadnic, kde jsou jednotlivé osy navzájem kolmé. Tento systém nám tedy umožňuje pohybovat se kdekoliv v prostoru, ovšem v každém bodě se můžeme nacházet pouze jednou možnou kombinací os.

## 3 PŘÍSTROJOVÉ VYBAVENÍ

### 3.1 PROGRAMOVATELNÝ AUTOMAT

#### 3.1.1 Šasi 1756-A7 se zdrojem [19]

Toto šasi je určeno pro řadu ControlLogix, může být osazeno dvěma druhy zdrojů, a to 1756-PA72 nebo 1756PB72. Naše šasi je vybaveno zdrojem 1756-PA72, obsahuje 7 slotů pro umístění libovolných modulů.



Obrázek 3.1 Šasi 1756-A7 se zdrojem

#### 3.1.2 Procesor 1756 – L62S [18]



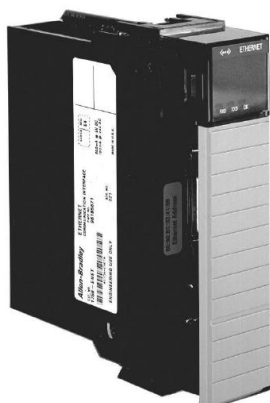
Obrázek 3.2 Procesor L62S vlevo a LPS vpravo

Jedná se o procesor řady GuardLogix, tedy procesor, který zajišťuje bezpečné ovládání. Systém GuardLogix se skládá z procesoru řady ControlLogix, který je spřažen s bezpečnostní částí 1756-LPS. Výhodou tohoto systému je, že tvoří celek, který v sobě zahrnuje standardní i bezpečnostní část. Primární procesor (1756-L62S) zpracovává standardní a bezpečnostní funkce a komunikuje s bezpečnostním

procesorem (1756-LPS). Ten poskytuje primárnímu procesoru redundanci pro bezpečnost. Programová paměť procesoru 1756-L62S je 4MB pro standardní část a 1MB pro bezpečnostní část.



### 3.1.3 Modul EtherNet/IP 1756-ENBT/A



Obrázek 3.3 Modul EtherNet/IP 1756-ENBT/A

Tento modul umožňuje komunikaci s ostatními zařízeními pomocí sítě EtherNet/IP, což je průmyslová síť, která vychází ze sítě ethernet s protokolem TCP/IP a jeho součástí je i protokol CIP (Common Industry Protokol), který zajišťuje bezpečnost a reálnou dobu odezvy, a využívá se i u sítí ControlNet a DeviceNet. U sítě EtherNet/IP neznamena zkratka IP Internet Protokol, což je lehce matoucí a mohlo by to k tomu zavádět, ale jde o Industrial Protokol.

### 3.1.4 Modul DeviceNet 1756-DNB



Obrázek 3.4 Modul DeviceNet 1756-DNB  
[20]

Jde o další komunikační modul. Komunikuje s rozhraními DeviceNet sítě, která byla vyvinuta americkou firmou Allen-Bradley. Aby se tato síť rychle rozšířila, byl uvolněn její standard. Je založena na sítích CAN (Controller Area Network), které byli původně vyvinuty pro automobilový průmysl. Maximální délka vedení je 500 m a připojit se může až 64 zařízení. V našem případě komunikuje s DeviceNet safety modulem I/O 1971DS-IB8XOB8.

### 3.1.5 Modul AS-i 3.0 Master/Scanner



Obrázek 3.5 Modul AS-i 3.0  
Master

Jedná se o modul firmy Bihl+Wiedemann pro řadu ControlLogix, který obsahuje 2 mastery, takže lze připojit 2 sítě AS-i. Tento modul zajišťuje pouze komunikaci po sběrnici AS-Interface, napájení této sběrnice musí být zajištěno zvlášť. Jelikož modul není od firmy Rockwell Automation, automat jej rozpoznává jako všeobecný modul. Dá se přes něj přiřazovat adresy modulům, a to buď přímo pomocí tlačítek a displeje, nebo pomocí přiloženého softwaru „AS-i ContolTool“ přes sériovou linku RS232. To však nebylo nutné, jelikož slavy již měli adresy přiřazeny.

### 3.1.6 Procesor 1756-L73

Jde o standardní procesor řady ControlLogix s programovou pamětí 8MB. Slouží k řízení modelu Labyrintu, protože pro řízení obou modelů užívám procesory zvlášť.

### 3.1.7 Modul SERCOS interface



Obrázek 3.6 Modul  
SERCOS interface

Komunikační systém SERCOS (SErial Real-time COmmunications System) je určen pro řízení digitálních servopohonů Allen Bradley. Je u něj kladen důraz na komunikaci v reálném čase a díky tomu, že je realizována pomocí optických vláken, zaručuje vysokou odolnost vůči rušení. Při použití plastických skleněných vláken může vedení dosahovat až 32 metrů, a při použití skleněných vláken se tato vzdálenost prodlužuje až na 200 metrů. Samozřejmostí tohoto vedení je odolnost vůči elektromagnetickému rušení. Systém Sercos je určen ke spolupráci s pohony Allen Bradley a moduly ControlLogix, takže je zajištěna snadná konfigurace, programovatelnost a monitorování pomocí standardního programního rozhraní

RSLogix 5000, ve kterém je možno využít až 32 instrukcí pro tento systém v dostupných programovacích jazycích. Další výhodou je možnost připojení zařízení za chodu a jeho výměna bez konfigurace. V systému je zabudována i diagnostika, která informuje řídicí systém o stavu pohonu. V sestavě automatu je použit modul 1756-M08SE, který umožňuje řídit až 8 os.

## 3.2ŘÍZENÍ

### 3.2.1 Kinetix 2000 [17]

Platforma Kinetix obsahuje několik typů měničů, které se od sebe liší výkonem, které jsou schopné ovládat. V tomto případě se jedná o nejslabší systém řízení servomotorů firmy Allen-Bradley z řady Kinetix. Maximální přípustný výkon připojeného motoru je do 3kW. Měniče Kinetix 2000 komunikují s PLC pomocí sítě SERCOS optickými kabely. Systém je optimalizován pro řídicí systémy ControlLogix a zjednodušuje tak konfiguraci přímo pomocí prostředí RSLogix 5000. Kinetix 2000 je modulární systém- první servoměnič obsahuje interface Sercos pro komunikaci a napájecí zdroj, pomocí kterého se napájejí ostatní měniče, přidané do následujících slotů.



Obrázek 3.7 Kinetix 2000

### 3.2.2 Servomotory řady MPL



Obrázek 3.8 Motor MPL A310

Servomotory řady MPL (MP Low Inertial Servo Motors) jsou střídavé bezkartáčové motory se zpětnou vazbou, kterou zajišťuje integrovaný enkodér. Vyznačují se nízkou setrvačností, způsobenou jejich sníženou hmotností. Tělo motoru je vyrobeno z nerezové oceli, aby se mohl používat i v potravinářském průmyslu, a hřídel je kalená pro přenos vyššího momentu. Tato řada byla vyvinuta pro plnou kompatibilitu se systémem Allen-Bradley Kinetix. Motory použité na modelu manipulátoru jsou napájeny napětím 230V, dosahují maximálních otáček 3000 ot/min, a jsou vybaveny inkrementální zpětnou vazbou. Pro osy X a Y jsou použity servomotory MPL A310F HK22AA, pro osu Z pak motor MPL A310 HK24AA, vybaven 24VDC brzdou pro dobu nečinnosti motoru.

### 3.3 MODEL LABYRINTU

Model labyrintu vznikl již v minulém akademickém roce jako bakalářská práce studenta Petra Zbranka. Já jsem tedy dostal hotový přípravek pro práci, který bylo potřeba propojit s PLC Allan-Bradley firmy Rockwell Automation. Musel jsem zajistit komunikaci modelu pomocí master modulu firmy Bihl+Wiedemann zapojeného do PLC.

### 3.4 Konstrukce

Rám konstrukce modelu labyrintu se skládá z Aluprofil 40 x 80 L, má čtvercový tvar, jehož vnitřní rozměr je 30x30cm. Vnitřní prostor rámu vyplňuje plochá železná deska, na které je černou izolační páskou vyznačeno rozdělení polí. Polí je na modelu 9, tedy při čtvercovém tvaru 3x3 polí. V každém poli jsou vyvrtány dva otvory, do kterých jsou vsazeny po jedné červená a zelená dioda. Ty slouží k navigaci v labyrintu. Na jedné straně, zvolené jako přední, je rám nastaven ještě o jeden kratší profil navíc, sloužící jako vstup do labyrintu. Z vnější strany profilu rámu jsou připevněny DIN lišty pro upevnění slavnů a zesilovačů řídicí elektroniky labyrintu. Na horní části rámu a vstupu jsou namontovány uchycovací úhelníky, na nichž jsou přidělané snímače, sloužící jako světelné závory pro rozpoznání přechodu mezi poli labyrintu. Na vstupu pak dva snímače proti odrazce, sloužící k rozpoznání vstupu nebo výstupu z labyrintu.

Na DIN liště jsou pak tedy rozmístěny moduly typu slave pro komunikaci s PLC následovně. Na levé straně modul digitálních vstupů a šestkrát vyhodnocovací systém pro světelné závory, na straně pravé pak pět modulů digitálních výstupů. Moduly dvou tlačítek a diod pro jejich podsvícení jsou pak umístěny ze přední strany pro ovládání labyrintu. Napájení AS-I a spínaný zdroj pro pomocné napájení jsou pak umístěny mimo model na samostatné DIN liště.

Kabeláž je modelem pokud možno vedena v profilu, nebo tak aby byla co nejvíce skryta.

Přesnou velikost a přesné rozmístění použitých součástek můžeme vidět v příloze 2.

### 3.5 Komponenty

Na modelu je tedy potřeba řídit 20 led diod z toho 18 navigačních a 2 v tlačítkovém modulu. Dále je potřeba vyhodnocovat 8 vstupních signálů, přičemž 6 je ze světelných závor průchodu labyrintem a již zmíněná 2 tlačítka. Proto jsou na modelu tyto komponenty:

AC5213, IFM Electronic, slave – Digitální výstupy

AC5210A, IFM Electronic, slave – Digitální vstupy

AC2088, IFM Electronic, A/B-Slave - aktivní horní část modulu AS-i modul světelný snímač

AC5000, IFM Electronic, Spodní modulový díl FK

OV5012, IFM Electronic, Vyhodnocovací jednotka pro optické senzory

OE0004, IFM Electronic, Reflexní světelná závora

OE0003, IFM Electronic, Jednocestná světelná závora, Vysílač

OE0002, IFM Electronic, Jednocestná světelná závora, Přijímač

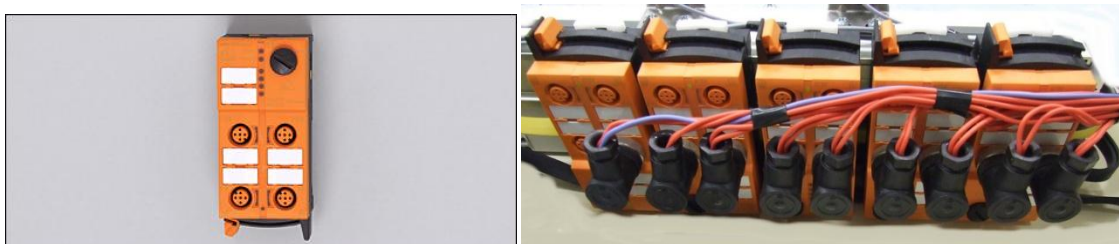
Dále pro napájení umístěny na zvláštní DIN liště:

- AS-i POWER, Siemens, napájení sběrnice AS-Interface
- JS-30-240/DIN, BKE, spínaný napájecí zdroj 24V/30W pro pomocné napájení

PLC a modul AS-Interface byli použity následující:

- ControlLogix L73, Allen-Bradley, PLC
- AS-Interface 3.0 Master/Scanner, Bihl+Wiedemann, master modul pro ControlLogix

### 3.5.1 Slave AC 5213 [8]



Obrázek 3.9 AC5213 a) vlevo – samostatný modul b) vpravo – osazení na modelu

Aktivní ClassicLine Modul, propojitelný pomocí AS-Interface plochého kabelu. Slouží k řízení digitálních výstupů, v našem případě tedy výstupní led diody pro navigaci. Na modelu je osazen 6x a každý modul ovládá 4 bity (diody). Každý z těchto slavů zabírá 1 adresu.

### 3.5.2 Slave AC5210 [9]



Obrázek 3.10 AC5210 a) vlevo – samostatný modul b) vpravo – osazení na modelu

Slouží k vyhodnocování vstupních dat, v našem případě přijímá signály od vyhodnocovacího systému, resp. od 6 snímačů (světelných závor), a přeposílá informace masteru. Ačkoli je tento modul jen jeden, tak každý modul může nastavovat pouze 4 bity, jak již bylo zmíněno dříve. Proto tento modul zabírá dvě adresy.

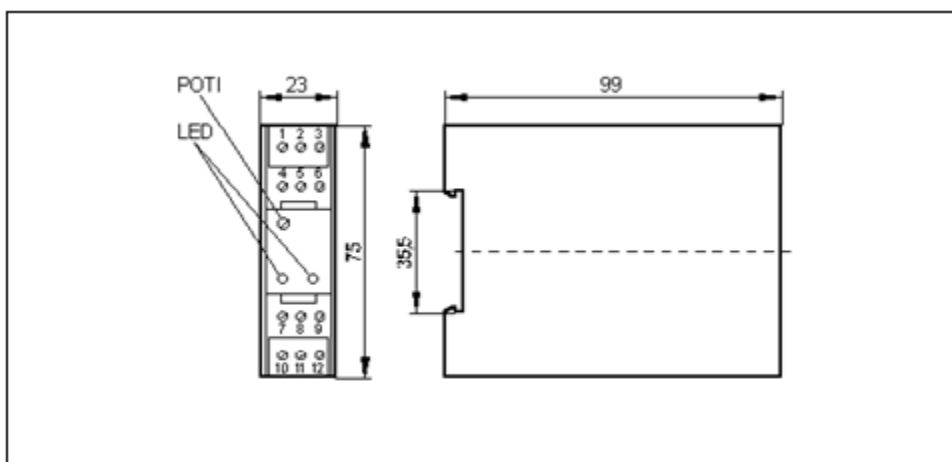
### 3.5.3 AC2088 a AC5000 [10],[11]



Obrázek 3.11 a) vlevo – AC2088 b) vpravo – AC5000

Tlačítkový modul s podsvícením tlačítek led diodami. Jde tedy o dva vstupy a dva výstupy. Společně tvoří adresně jeden slave. Tento modul slouží k signalizaci možnosti či nemožnosti vstupu do labyrintu. Tlačítka pak ovládáme spuštěním algoritmu navigace v labyrintu nebo spuštěním vyvedení člověka (panáčka) ven.

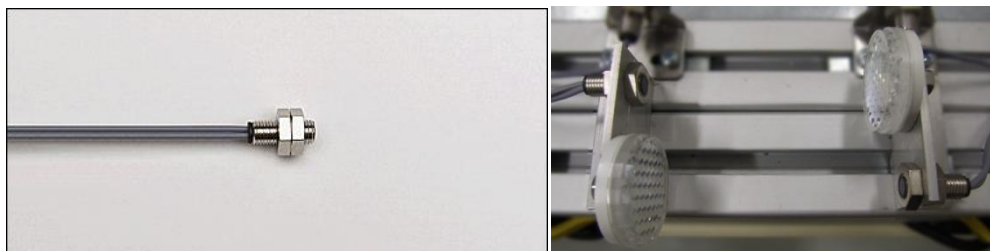
### 3.5.4 Vyhodnocovací systém OV5012 [14]



Obrázek 3.12 Vyhodnocovací systém OV5012

Slouží k vyhodnocování hodnot ze světelných závor. Na modelu je tedy osazen 6x. Nezabírá žádnou adresu, protože předává pouze binární hodnotu modulu vstupů AC5210.

### 3.5.5 Snímač OE0004 [12]



Obrázek 3.13 OE0004 a) vlevo – samostatné čidlo b) vpravo – osazení na modelu

Ve spojení se spínacím zesilovačem zjišťuje přístroj bezdotykově předměty. Lze ho použít jako reflexní světelnou závorku nebo jako světelný snímač. Osazen je proti odrazné ploše dvakrát na vstupu pro zjištění, zda se člověk pohybuje směrem do nebo z labyrintu.

### 3.5.6 Snímač OE0003 a OE0002 [13]



Obrázek 3.14 OE0003 a) vlevo – samostatné čidlo b) vpravo – osazení na modelu

Snímače sloužící jako světelná závorka. OE0003 jako vysílač a OE0002 jako přijímač. Opět zapojeny do zesilovače OV5012, sloužící k rozpoznávání přechodu mezi poli labyrintu. Technické parametry jsou obdobné jako u předchozího typu.



### 3.5.7 Napájení

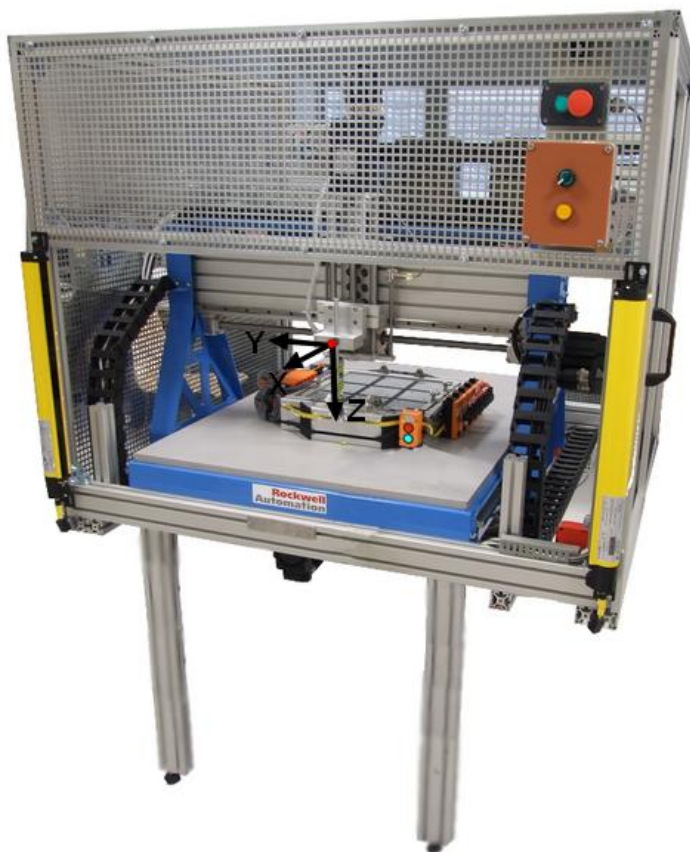


Obrázek 3.15 a) vlevo – pomocné napájení 24V spínaný zdroj BKE b) vpravo – napájení AS-I

Takto pak vypadají oba napájecí moduly na zvláštní DIN liště mimo model labyrintu. Oba jsou připojeny do sítě přes trojžilový kabel. Zajišťují napájení pro AS-I sběrnici (žlutý) a pomocné napájení pro slavy AS-I (černý).

### 3.6 MODEL MANIPULÁTORU

Model manipulátoru byl vyroben firmami Robotech a Zámečnictví – MSK, jeho přístrojové vybavení doplnil Bc. Štefan Golis, v rámci své bakalářské práce s názvem Řízení modelu manipulátoru. V tomto roce byl manipulátor vybaven dalšími prvky zabezpečení, doplněn o vizualizaci na PanelView Plus 600 pro tuto bezpečnost na a po celém obvodu zakryt plechovou mříží jako diplomová práce Bc. Ondřeje Peňáze.



Obrázek 3.16 Model manipulátoru

Nyní tedy bezpečnostními prvky jsou optická závora, nouzový vypínač vzadu na skříni řízení, bezpečnostní vypínač vpředu, přepínač pro přechod do servisního módu a s ním spjatý bezpečnostní zámek bočních dveří pro přístup k manipulátoru při servisním módu.

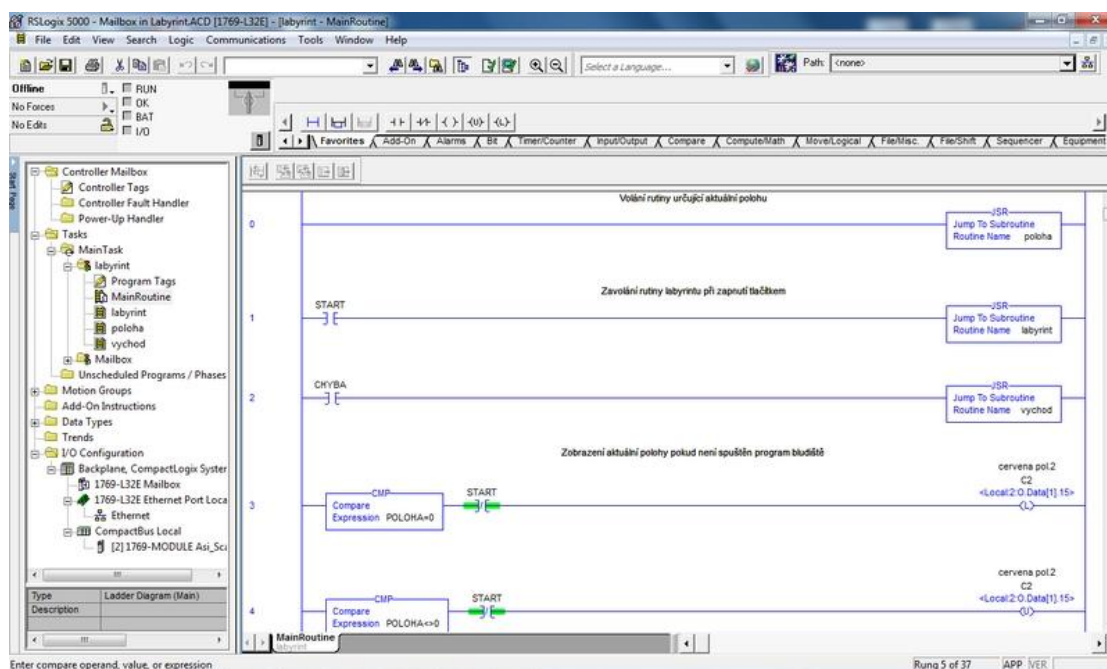
### 3.7 I/O modul 1971DS-IB8XOB8

Jde o Safety I/O modul, který má 8 digitálních vstupů a 8 digitálních výstupů. Tento modul složí k zabezpečení technologie před nenadálými událostmi. Jsou k němu tedy připojeny již zmiňované bezpečnostní prvky a další.

## 4 PROGRAMOVÁNÍ

Program byl vytvořen v programovacím prostředí RSLogix 5000, který je určený přímo pro automat Allen-Bradley, o kterém se dozvíme více v následujících kapitolách.

### 4.1 PROSTŘEDÍ PROGRAMU RS LOGIX 5000



Obrázek 4.1 Prostředí RS Logix 5000

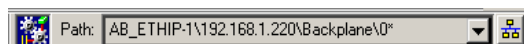
#### 4.1.1 Založení projektu

Pro zakládání projektů slouží program RSLogix 5000. Založení projektu je jednoduché a probíhá v několika krocích. Nejprve je tedy potřeba vytvořit projekt. K tomu slouží položka **File** → **New Project**. Poté musíme v okně *New Controller* zvolit typ a jeho revizi, ty se dozvíme v programu „RS Linx“. Dále pojmenujeme projekt a zadáme cestu, kam jej ukládat. Tím je samostatný projekt založen a zobrazí se pracovní obrazovka.

Následně je potřeba vybrat automat a zajistit komunikaci s ním. Můžete to udělat pomocí tlačítka z lišty nebo pomocí položky **Communications** → **Who Active**. Pro přístup z kteréhokoliv místa v síti k automatu zvolíme komunikaci přes Ethernet/IP vybráním automatu z ovladače **AB\_ETHIP-1, Ethernet**. Správný automat poznáte podle IP adresy. Dále je nezbytné přidat do projektu všechny potřebné periferní karty (moduly), které budeme potřebovat, a se kterými budeme komunikovat. Opět je potřeba

nastavit revizi modulu a také jeho pozici na automatu, kterou také zjistíme v programu „RS Linx“.

## 4.1.2 Propojení s PLC



Obrázek 4.2 Propojení s PLC

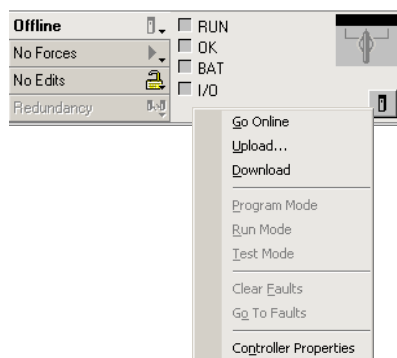
V tomto panelu nastavujeme aktuální cestu k PLC. Propojení probíhá pomocí programu RS Linx, který zajišťuje komunikaci s PC.

## 4.1.3 Struktura projektu

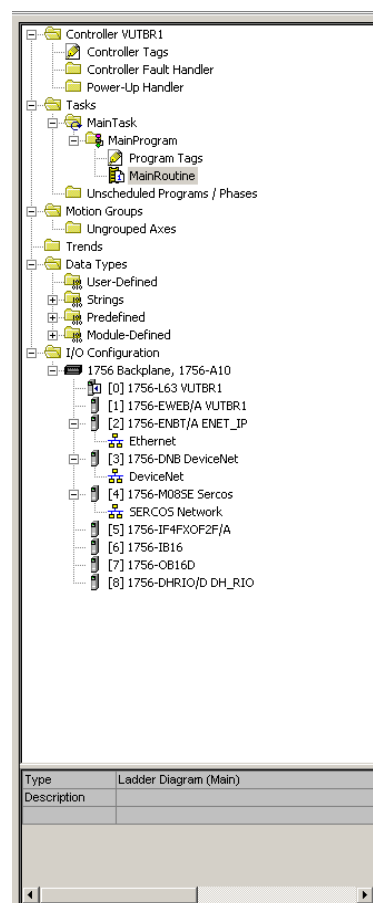
V levém panelu můžeme vidět strukturu PLC a všechny dostupné programy a tagy controleru. Jedná se tedy o strukturu celého projektu.

## 4.1.4 Online panel

Slouží ke změně režimu PLC a k uploadu a downloadu programu.



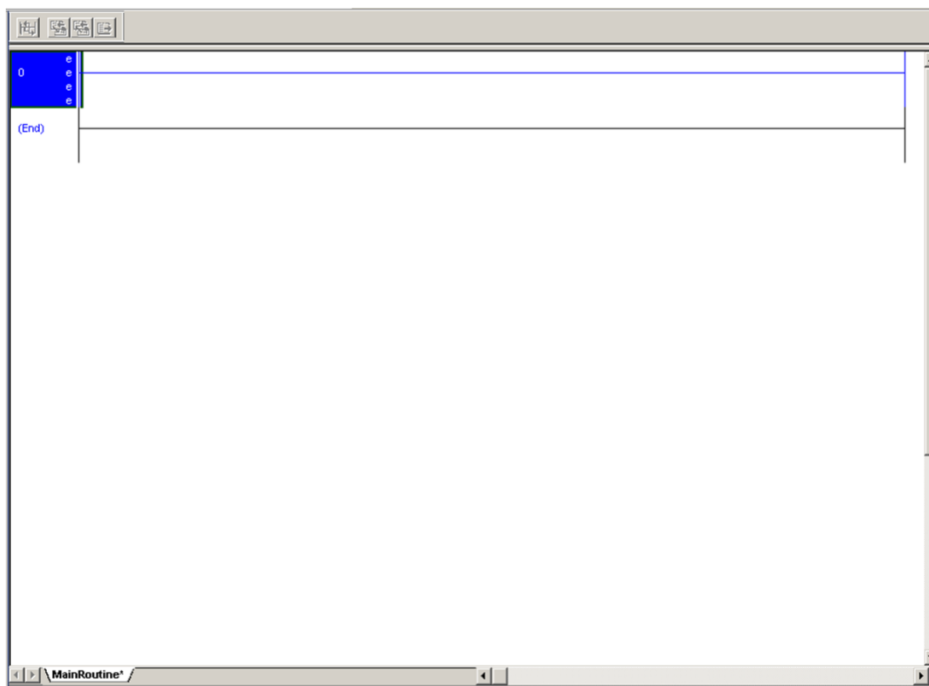
Obrázek 4.3 Online panel



Obrázek 4.4 Struktura projektu

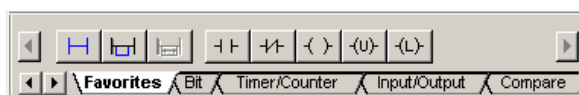
### 4.1.5 Okno hlavního programu

V tomto okně se vytváří samotný program nebo vytváří či mění tagy. Okno vždy vypadá jinak dle zvoleného jazyka programování.



Obrázek 4.5 Okno hlavního programu

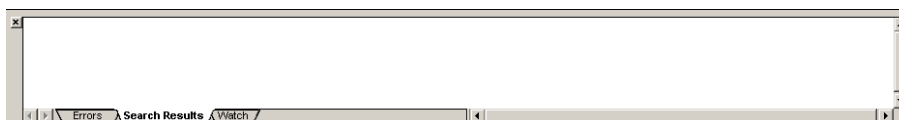
### 4.1.6 Panel nástrojů prvků zvoleného programovacího jazyka



Obrázek 4.6 Panel nástrojů LAD

Z tohoto panelu vybíráme logické, matematické či časové funkce, které používáme v programu.

### 4.1.7 Informační okno



Obrázek 4.7 Informační okno

Informuje o chybách a výsledcích programu, kompilace či samotné komunikace.

### **4.1.8 Definice proměnných (tagů)**

Nedílnou součástí programu je nadefinování proměnných a jejich typů. Také pro větší přehled vstupů a výstupů je potřeba tyto nadefinovat jako vnitřní proměnné, a přiřadit je jako aliasy, zejména nám to pomůže v přehlednosti programu. Proměnným musíme vždy přiřadit typ nebo ho přiřazuje alias pokud jej proměnná má. Nejpoužívanějším typem je binární BOOL. Dále je potřeba nadefinovat proměnnou pro každý časovač či čítač, typy se nazývají TIMER a COUNTER.

### **4.1.9 Programovací jazyk**

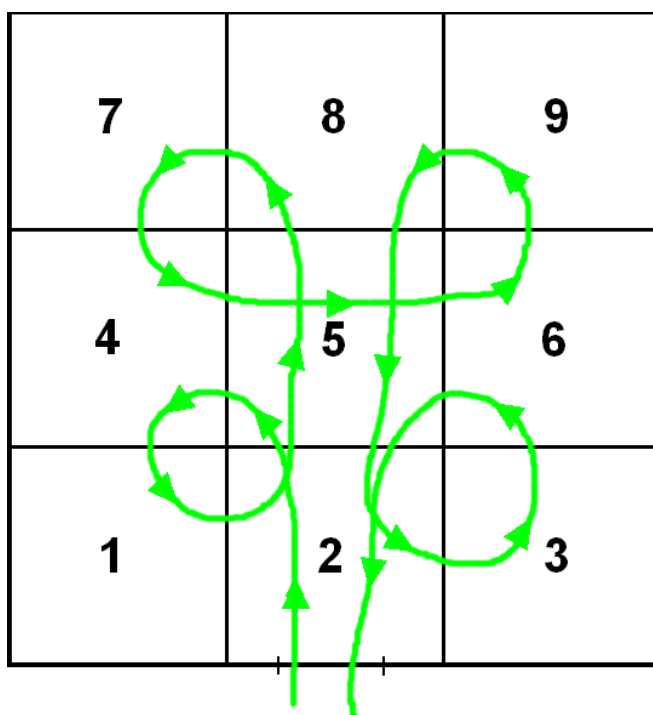
Jako programovací jazyk jsem volil LAD (Ladder Diagram), česky žebříčkový diagram. Tento jazyk jsem si vybral pro jeho jednoduchost a transparentnost. Pokud programátor popíše jednotlivé latky, je program lehce čitelný, a neměl by být velký problém s modifikací třeba i jiným programátorem.

## 4.2 PROGRAM MODELU LABYRINTU

Program zajišťuje několik funkcí. V první řadě zjišťuje, zda jsou všechny slavy připojeny a pracují správně. Dále zjišťuje aktuální polohu člověka v bludišti, a pokud není spuštěn program průchodu labyrintem a člověk se jím pohybuje pouze volně, zobrazuje jeho aktuální polohu rozsvícením zelené diody políčka, na kterém právě stojí. Při spuštění programu průchodu labyrintem s navigací program ukazuje zelenou diodou vždy další krok pro správný průchod labyrintem, přičemž lze volit ze dvou cest, jelikož vchod a východ je stejný. Cestu můžeme vidět na obrázku *Obr. 3.1 Cesta průchodu*, přičemž první cesta vede po směru vyznačených šipek a druhá pak proti směru těchto šipek. Tedy možné cesty jsou následující:

$$\rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 7 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 8 \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 2 \rightarrow$$

nebo v opačném směru

$$\rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 7 \rightarrow 8 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow$$


### Obrázek 4.8 Cesta průchodu

Poslední funkcí, kterou je labyrint vybaven, je vyvedení člověka ven. To je započato, pokud člověk vstoupí na špatné políčko při procházení labyrintem. Pak je naveden na střed a tudy ven z labyrintu. Program samozřejmě řeší více dílčích úkolů, kterou jsou popsány v dalších kapitolách.

## 4.2.1 Přidání modulu AS-I

Jak jsem již zmínil, nebylo možné přidat modul klasickým způsobem, tak aby byla zajištěna bezchybná komunikace, a přístup ke slavům a jejich bitům. Bylo tedy potřeba stáhnout ukázkový projekt ze stránek výrobce, a z něj vycházet. Takže v tomto projektu je potřeba změnit Controller na správný typ a revizi. Dále je také potřeba nastavit cestu jak již bylo zmíněno. V poslední řadě pak pozměnit pozici, na které se modul nachází. V našem případě je umístěn na pozici 2 pod sběrnici CompactBus. Tímto způsobem je zajištěna komunikace po AS-I sběrnici se slavy.

## 4.2.2 Přístup ke slavům

Přístup k jednotlivým slavům a ovládání je popsáno v manuálu modulu opět volně stažitelného ze stránek výrobce. Vše je funkční díky vloženému programu nazvanému *Mailbox*, který zajišťuje patřičnou komunikaci Controlleru s modulem AS-I a následně přístup k jednotlivým bitům slavů. Tím máme tedy přístup k několika skupinám tagů, jak můžeme vidět na obrázku *Obr. 3.2 ControlTags*. Můžeme tedy přistupovat k bitům několika způsoby. Zvolil jsem nejjednodušší a již předem známý, přes skupinu *Local*, kterou vyjadřuje typ INT a jednotlivé bity jsou tedy typu BOOL. Pak už jen stačí najít patřičné bity dle adresy slavu pomocí tabulky v manuálu Scanneru AS-Interface.

Scope: Mailbox Shgw... Show All					
	Name	Value	Force Mask	Style	Data Type
-	ASi_Input	{ ... }	{ ... }		AB:1769_MODULE_INT_72Bytes:I:0
+	ASi_Input.Fault	0		Decimal	DINT
+	ASi_Input.Data	{ ... }	{ ... }	Decimal	INT[34]
-	ASi_Output	{ ... }	{ ... }		AB:1769_MODULE_INT_68Bytes:O:0
+	ASi_Output.Data	{ ... }	{ ... }	Decimal	INT[34]
+	Local2:C	{ ... }	{ ... }		AB:1769_MODULE:C:0
+	Local2:I	{ ... }	{ ... }		AB:1769_MODULE_INT_72Bytes:I:0
+	Local2:O	{ ... }	{ ... }		AB:1769_MODULE_INT_68Bytes:O:0
-	Mailbox1	{ ... }	{ ... }		Mailbox
+	Mailbox1.Request	{ ... }	{ ... }	Hex	SINT[36]
	Mailbox1.Start	0		Decimal	BOOL
+	Mailbox1.Response	{ ... }	{ ... }	Hex	SINT[36]

Obrázek 4.9 ControlTags

## 4.2.3 Použité funkce

V programu výhradně používám kontakty XIC a XIO, cívky OTL a OTU. Z matematických funkcí všeobecný komparátor CMP, z logických hlavně funkci MOV pro přiřazení hodnoty proměnné, samozřejmostí jsou i časovače TOF. Využívám i několik speciálních funkcí, například pro práci se subrutinami.

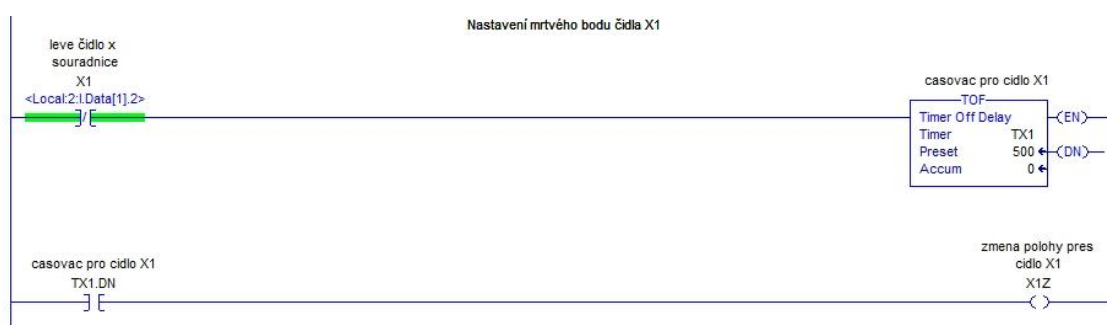


## 4.2.4 Vlastní program labyrintu

Program se skládá ze čtyř rutin, přičemž jedna je hlavní a tři jsou subrutiny, které jsou volány funkcemi z hlavní rutiny. Hlavní rutina *MainRoutine* tedy rozhoduje o běhu všech ostatních rutin. Dále zajišťuje všechny úkony, které se dějí mimo program navigace. Tedy se stará o indikaci polohy člověka v labyrintu, pokud není spuštěn program navigace. Vždy, když se tomu tak stane, rozsvítí se zelená led dioda na políčku, na kterém se člověk právě nachází. Zároveň se v tuto chvíli rozblíká červené tlačítko na vstupu, aby do labyrintu nevstupovala žádná další osoba. Pokud je poloha rovna nule, tedy není-li nikdo v labyrintu, na vstupním semaforu svítí zelená dioda a tato poloha je indikována rozsvícením červené diody na poli 2 za vstupem. Hlavní rutina tedy výhradně zajišťuje signalizaci labyrintu, spouští nebo vypíná program navigace a s tím volá další potřebné subrutiny. V této rutině jsou i naprogramovány některé chybové situace, které mohou nastat při chodu programu, ale i ty, které vzniknou třeba špatnou komunikací se slavy.

### 4.2.4.1 Určení polohy

K určování polohy slouží subrutina *poloha*. Pracuje se světelnými závorami, díky kterým pak mění aktuální polohu člověka v programu, a to změnou INT tagu *poloha* pomocí logické funkce MOV. Jelikož hodnoty z čidla nejsou jednoznačné, jak by se mohlo zdát, ale čidlo jako kdyby kmitá s hodnotami, je potřeba nastavit „mrtvou dobu“ čidla, kdy nebude bráno v potaz, jaká hodnota je na čidle. Vezmeme tedy první sestupnou hranu čidla, která sepne na krátkou dobu časovač, a tím vytvoříme prostor pro průchod přes čidlo, aniž by se přejití započítalo několikanásobně. Řešení je jednoduché a můžeme jej vidět na obrázku *Obr.4.10 Mrtvá doba čidla*.



Obrázek 4.10 Mrtvá doba čidla

Rozpoznání vstupu nebo výstupu z labyrintu je jednoduché, jelikož u vstupu jsou dvě čidla za sebou, takže podle toho v jaké posloupnosti přijdou signály, poznáme, jakým směrem člověk jde. Při pohybu v labyrintu je však mezi každými poli jen jedna světelná závora, takže musíme vycházet z předpokladu předešlého pole.

#### 4.2.4.2 Navigace labyrintem

Při spuštění programu navigace je spuštěna subrutina *labyrint*, která zajišťuje navigaci a kontrolu správného projití dle sekvence. Program pracuje na porovnávání dvou hodnot, a to hodnoty čísla kroku, který se s každým posunutím navyšuje, s polohou osoby v labyrintu. Pokud je poloha správná vůči kroku, program rozsvítí další zelenou diodu navigačního procesu. Projití labyrintem je tedy rozděleno na 20 již dříve zmíněných po sobě jdoucích kroků. V kroku 2, kdy je osoba na políčku 5, má možnost rozhodnout, zda půjde labyrintem doleva nebo doprava. Tedy jestli projde labyrint po nebo proti směru hodinových ručiček. Pokud se rozhodne jít po směru, nastaví se tag *CESTA2*, aby bylo možné odlišit, zda je na správném políčku. Trasa labyrintu jde vidět již v dřívější kapitole na obrázku *Obr. 3.1 Cesta průchodu*. Po celou dobu běhu tohoto programu bliká zelený semafor na vstupu, který indikuje, že právě probíhá program navigace, a zároveň svítí červený semafor, aby do labyrintu nevstupovala další osoba.

V případě, že osoba nejde podle navigace a přejde na jiné pole, program nastaví tag *CHYBA* na hodnotu 1.

#### 4.2.4.3 Vyvedení z labyrintu

V případě, že je hodnota tagu *CHYBA* nastavena na 1 a zároveň je spuštěn program navigace, tak je spuštěna rutina *východ* a rutina *labyrint* je přerušena. Vyvedení lze také zapnout stiskem červeného tlačítka semaforu na vstupu, čímž se přeruší navigace a zapne se vyvedení z labyrintu. Při chodu této rutiny blikají všechny červené diody v labyrintu i semafor na vstupu. Vyvedení opět používá polohu osoby a naviguje ji tak, aby vyšla z labyrintu co nejkratší cestou. A to tak, že nejprve na střed, pokud se nachází v krajních polích, a pak středem ven. Po východu ven je opět nastaven tag *CHYBA* na hodnotu 0, čímž je vyvedení ukončeno a labyrint je opět v počátečním stavu.

#### 4.2.4.4 Chybové stavy

Tyto stavy nebo situace nastavují proměnnou *CHYBA* na hodnotu 1 (true):

- pokud osoba při navigaci vstoupí na jiné pole, než ji navádí program,
- pokud je po dobu běhu programu navigace zmáčknuto červené tlačítko na vstupu,
- pokud je v labyrintu osoba, a pokusí se do něj vstoupit druhá

#### 4.2.4.5 Komunikace s manipulátorem

Pro předávání stavů labyrintu byla vytvořena nová subrutina *labyrint\_out* a proměnná *LABYRINTOUT* datového typu *dint*, což je v podstatě pole 32 proměnných typu *bool*, na něž byly ve zmíněné rutině navázány potřebné proměnné pro manipulátor. Dále byl typ této proměnné nastaven jako *Produced* pro zajištění komunikace s procesorem manipulátoru. O zajištění této komunikace se více dozvíme v kapitole 4.3.1.1 .

## 4.3 PROGRAM MODELU MANIPULÁTORU

Tento program zajišťuje pohyb manipulátoru. Součástí programu je bezpečnostní část, která v případě zásahu do bezpečnosti stroje vypíná jednotlivé osy či celé řízení, tak aby byla ochráněna obsluha stroje. Bezpečnostní část programu, a s ním spjatá část programu vizualizace, je prací Bc. Ondřeje Peňáze, který mi tuto část poskytl. Více o zabezpečení se tedy můžeme dozvědět v jeho práci.

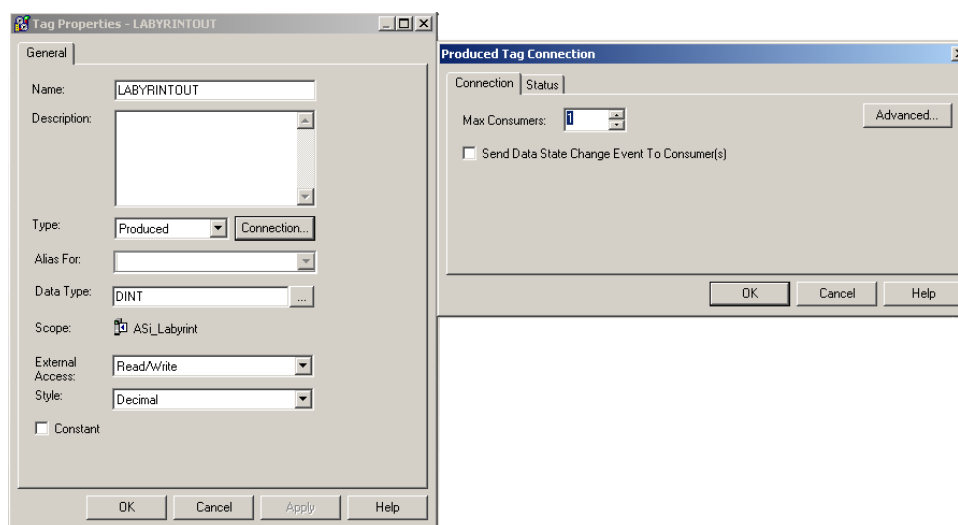
### 4.3.1 Konfigurace

Pro přístup ke všem prvkům modelu manipulátoru je potřeba přidat všechny moduly PLC, následně nakonfigurovat souřadný systém a přidat všechny osy pro řízení motorů. Tuto konfiguraci jsem však nemusel provádět, protože jsem ji pouze převzal z diplomové práce Bc. Pavla Kováře, na téma Model CNC frézky. Tato konfigurace mi byla poskytnuta vedoucím práce, já jsem se s ní pouze seznámil, jelikož vytvoření vlastní konfigurace by bylo časově náročné.

Mým úkol pro konfiguraci bylo zajistit komunikaci mezi modelem labyrintu a modelem manipulátoru, tak aby mohl být manipulátor řízen dle pokynů labyrintu.

#### 4.3.1.1 Zajištění komunikace mezi procesory

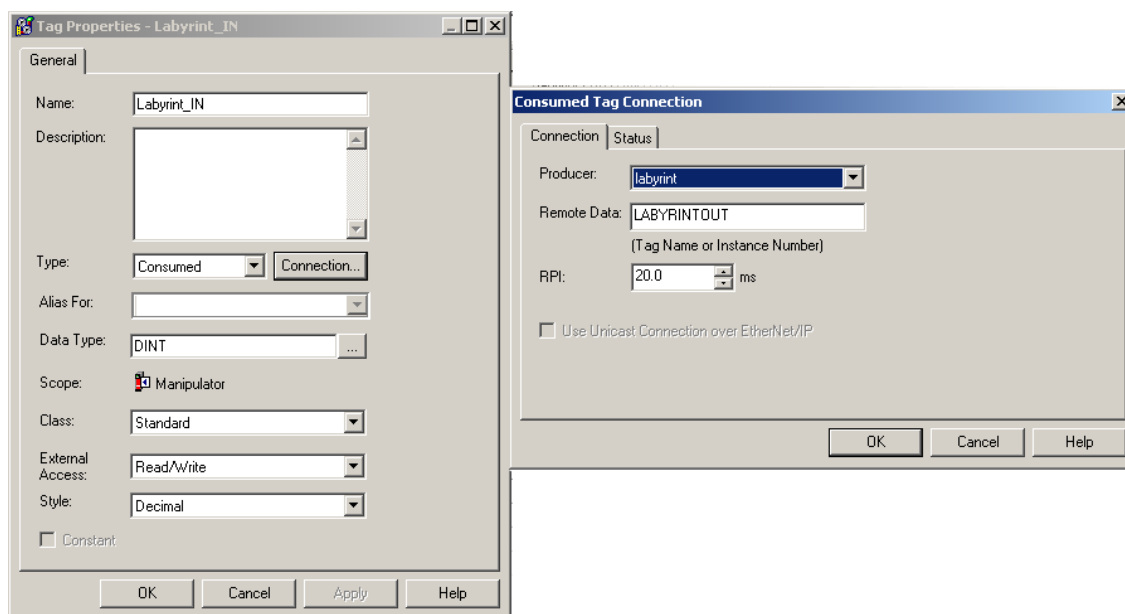
Bylo potřeba zajistit, aby manipulátoru byly předávány proměnné stavu labyrintu a proměnné potřebné k navigaci manipulátoru. Nejdříve jsem tedy vytvořil příslušnou proměnnou v programu labyrintu, nazývá se LABYRINTOUT. Typ této proměnné byl nastaven jako *Producer* a v rozšířeném nastavení *Connection...* se zvolí, kolika procesorům bude umožněno se na tuto proměnnou navázat, v tomto případě stačí hodnota 1.



Obrázek 4.11 Nastavení proměnné producenta

Pro přístup k této proměnné je potřeba do konfigurace programu manipulátoru přidat příslušný procesor, který tuto proměnnou předává. To učiníme přidáním nového modulu na Backpanel, kdy vybereme příslušný procesor, jeho pozici v šasi a jeho revizi. Revizi i pozici zjistíme pomocí komunikačního programu RS Linx.

Následně je potřeba vytvořit v programu manipulátoru proměnnou stejného datového typu jako byl *Produced*, tak aby mohly být tyto proměnné na sebe navázány. Pojmenoval jsem ji *Labyrinth\_IN* a následně bylo potřeba nastavit komunikaci. Typ pro příjem musí být nastaven jako *Consumed* a následně, v rozšířeném nastavení *Connection...*, vybrat procesor producenta a název proměnné pro navázání, v našem případě tedy *LABYRINTOUT*. Jako poslední možnost si můžeme zvolit, jak často se bude obnovovat hodnota této navázané proměnné. Ponechal jsem výchozí hodnotu 20ms.



Obrázek 4.12 Nastavení proměnné konsumenta

#### 4.3.1.2 Přístup k I/O safety bloku

Jelikož na modelu manipulátoru vykonávalo svou práci více studentů, bylo potřeba vždy převzít kontrolu nad I/O safety blokem, ke kterému jsou připojeny bezpečnostní prvky. Protože přístup je po síti DeviceNet, a jde o bezpečnostní prvek systému, může k němu mít přístup jen jedna konfigurace. Proto je potřeba převzít práva k tomuto safety blocku v jeho nastavení, tak aby komunikoval s naším aktivním projektem. To můžeme provést v nastavení safety blocku v záložce Safety, kde stiskneme tlačítko Reset Ownership, a tím se safety block přihlásí k naší aktivní konfiguraci a nastaví komunikaci pro náš projekt.

### 4.3.2 Použité funkce

Pro programování servopohonů využíváme několik Motion instrukcí. Díky konfiguraci souřadného systému je to velmi jednoduchý způsob programování pohonů, kde stačí nastavit pouze několik parametrů pro správný chod.

Výstupy instrukcí informují o stavu:

EN - nastaví se na logickou 1, když je na vstupu logická 1,

DN - nastaví se na logickou 1, když se instrukce dokončila a nenastala žádná chyba instrukce, dokončení však nastává i při ukončení instrukce, takže se nemusí provést celá, když obsahuje výstup PC,

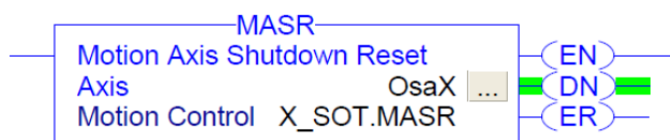
ER - nastaví se na logickou 1, když nastane chyba instrukce,

IP – nastaví se na logickou 1 při vykonávání instrukce,

PC – nastaví se na logickou 1 po dokončení instrukce, tedy po úplném provedení aniž by byla přerušena.

#### 4.3.2.1 MASR

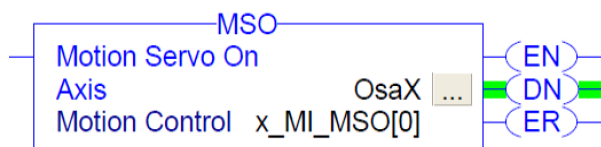
Motion Axis Shutdown Reset uvádí vypnutou osu zpět do stavu připravenosti k zapnutí. Parametry jsou osa (*Axis*), na které se má instrukce vykonat, a do parametru *Motion Control* zadáváme název proměnné typu *Motion instruction*, do které se ukládají informace o instrukci. Při vykonání této funkce jsou také smazány všechny poruchy spojené s touto osou.



Obrázek 4.13 instrukce MASR

#### 4.3.2.2 MSO

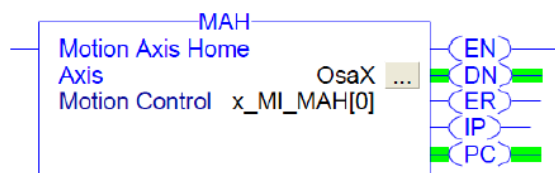
Motion Servo On aktivuje osu a připraví ji k používání. Většinou je dobré před touto instrukcí vykonat MASR, aby se na ose nenacházeli žádné chyby. Parametry této funkce jsou stejné jako u MASR, tedy osa a proměnná.



Obrázek 4.14 instrukce MSO

### 4.3.2.3 MAH

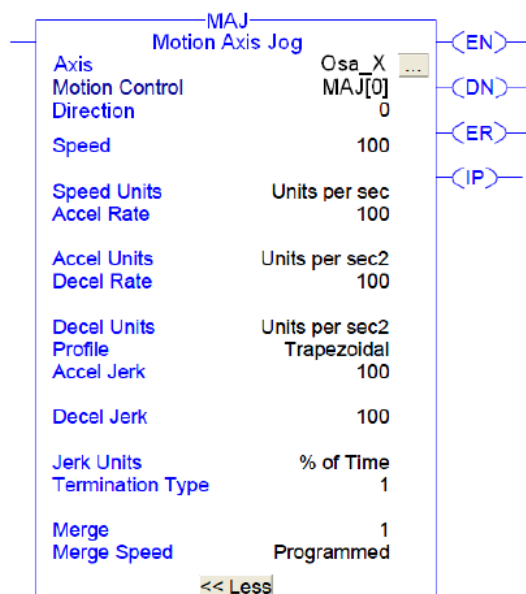
Motion Axis Home takzvaně zaparkuje zvolenou osu do výchozí polohy, na které se nachází snímač. Zároveň se provede inicializace souřadného systému osy. Tedy pokud by tato funkce nebyla po zapnutí spuštěna, nedošlo by ke kalibraci servomotoru a mohlo by dojít k havárii způsobené chybným hlášením o poloze. Parametry této funkce jsou stejné jako u předchozích.



Obrázek 4.15 instrukce MAH

### 4.3.2.4 MAJ

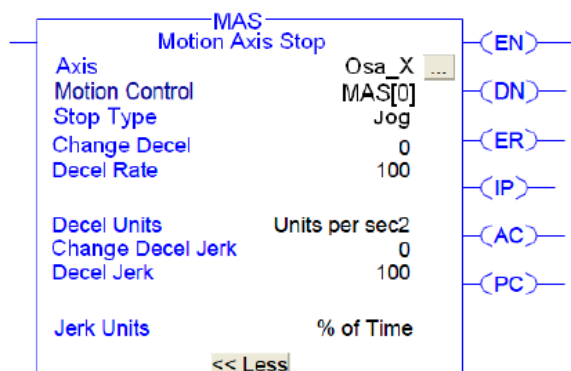
Motion Axis Jog uvádí danou osu do pohybu a nepřestane se vykonávat, dokud není tato instrukce přerušena jinou instrukcí, nejčastěji MAS. Parametry, které musí být instrukci dány je opět osa, kterou chceme pohybovat, směr jejího pohybu Direction (0 vpřed + a 1 vzad -), rychlost, zpomalení a zrychlení. Důležitým parametrem je také profil, podle kterého se bude zrychlovat a zpomalovat. Na výběr je z trapézového profilu nebo z S-křivky.



Obrázek 4.16 instrukce MAJ

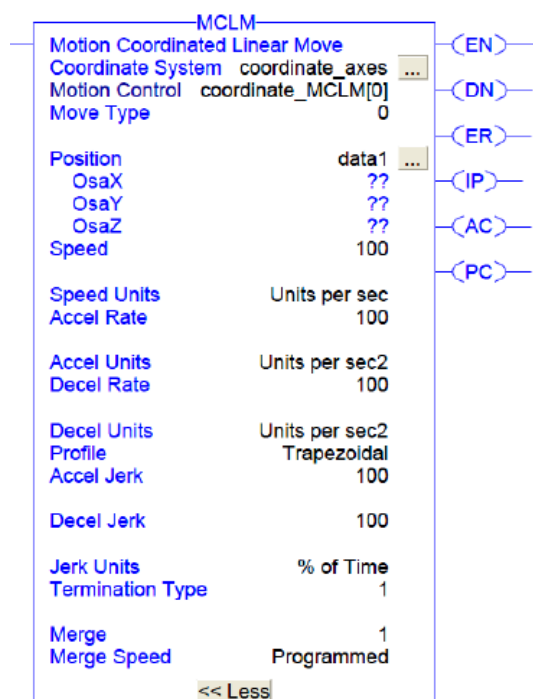
### 4.3.2.5 MAS

Jak již bylo zmíněno Motion Axis Stop slouží k zastavení pohybu osy. Kromě již zmíněných parametrů, je potřeba zadat parametr Stop Type, který označuje jaký pohyb, neboli kterou instrukci, bude zastavovat. Lze zastavovat všechny pohyby, nebo jen ten, který si tímto parametrem určíme. Dále je možnost nastavit parametr Change Decel, který udává zda má dojít k zastavení okamžitě nebo se zpomalením.



Obrázek 4.17 instrukce MAS

### 4.3.2.6 MCLM



Obrázek 4.18 instrukce MCLM

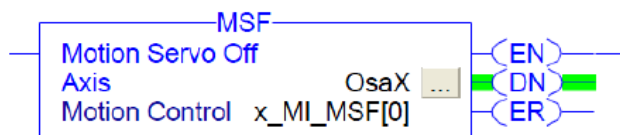
Motion Coordinated Linear Move slouží k lineárnímu pohybu v souřadném systému. Použitím instrukce dojde k lineárnímu přesunu osy z výchozího bodu do bodu zadaného v instrukci. Parametry této instrukce jsou odlišné než u předchozích. Prvním parametrem je *Coordinate System* tedy námi zvolený souřadný systém, *Move Type* udává, zda bude pohyb inkrementální (1) nebo absolutní (0). Tedy jestli bude konečná souřadnice brána jako absolutní v souřadném systému nebo jako inkrementální od momentální pozice. Parametr *Position* je pole, do kterého se zadávají hodnoty koncových pozic, kam se má systém pohnout. Dalšími parametry

jsou zrychlení a zpomalení a jako poslední parametr je *Termination Type*, který určuje, jakým způsobem bude instrukce ukončena a popřípadě navázána na další.



### 4.3.2.7 MSF

Motion Servo Off uvede osu do vypnutého stavu, osa setrvává v tomto stavu, dokud není opět zavolána instrukce MSO. Parametry i výstupy instrukce jsou stejné jako u předchozích.



Obrázek 4.19 instrukce MSF

## 4.3.3 Vlastní program manipulátoru

Program je rozdělen do několika rutin pro jeho obsáhlost a hlavně aby byl přehlednější a dalo se v něm snadněji orientovat, či provádět úpravy. Z hlavní rutiny jsou volány jednotlivé rutiny tehdy, když jsou potřeba a jsou splněny podmínky pro jejich volání.

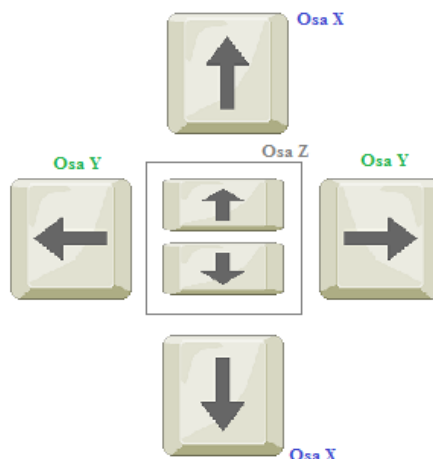
### 4.3.3.1 Inicializace

Tato rutina se provede při startu programu, nejdříve vymaže všechny nastavení proměnných na základní inicializační hodnoty. Poté čeká na spuštění řídicích měničů Kinetix a jakmile jsou spuštěny, tak po stisknutí resetovacího tlačítka připraví všechny osy k zapnutí (MASR), následně je zapne (MSO). Po zapnutí os a stisknutí tlačítka Hm ve vizualizaci nebo zeleného tlačítka na přední straně manipulátoru dojde k „houmování“ (MAH) všech os, nejprve však osa Z aby manipulátor případně do něčeho nenarazil a posléze až osy X a Y zároveň. Jakmile se provede celá tato inicializace v pořádku, nastaví se proměnná *initok* do logické 1, tím nás informuje i inicializaci souřadného systému a zpřístupní manuální ovládání manipulátoru.

### 4.3.3.2 Manuální ovládání

Tato rutina *pusuvJOG* dovoluje manuální ovládání všech os manipulátoru. Rutina však jen nespouští (MAJ) a nezastavuje (MAS) pojezd příslušných os, ale také hlídá takzvané softwarové limity, takže zabraňuje vyjetí až na koncová čidla manipulátoru. Vymezuje tedy celý souřadný systém. Hlídání softwarových limitů je vykonáváno pomocí matematických instrukcí LES a GRT, které reprezentují znaménkové funkce  $<$  a  $>$ .

Manuální ovládání v našem případě hlavně slouží k najetí před vstup do labyrintu pro následnou inicializaci polohy labyrintu v souřadném systému. Pro toto ovládání využíváme vizualizace, kde lze po inicializaci pomocí šipek ovládat jednotlivé osy.



Obrázek 4.20 Ovládací kříž pro manuální pojezd

#### 4.3.3.3 Inicializace labyrintu

Po nejetí před vstup do labyrintu a stisknutí tlačítka *Init Labyrint* ve vizualizaci se začne provádět tato rutina s názvem *initlab*. Zajišťuje zjištění, zda se před aktuální pozicí manipulátoru opravdu nachází vstup labyrintu. Rozjede se tedy směrem do labyrintu, a pokud do časového rozmezí, které udává časovač *TINITLAB*, nenajede na vstup labyrintu tak se zastaví a vyhodnotí, že zde se labyrint nenachází. V případě, že opravdu najede do vstupní brány labyrintu, pokračuje dále dovnitř, poté přejede na levé pole a zpět na pravé pole, tak aby projel přes obě čidla na šířce labyrintu. Obě pozice při přejetí čidla se zapíše a následně je vypočítána absolutní pozice pole za vstupem (pole označené číslem 2) a manipulátor najede do středu tohoto pole a vyjede ven z labyrintu. Tím je připraven pro pojezd v labyrintu. Zamezí používání manuálního ovládání, tudíž se již nelze pohnout v žádné ose, aby nebylo nutné znovu inicializovat labyrint. Naopak povolí používání pojezdu v labyrintu.

#### 4.3.3.4 Pojezd v labyrintu

Rutina *pohybLAB* zajišťuje pohyb v labyrintu po vykonání všech předchozích inicializačních kroků. Pohyb je vykonáván pomocí instrukce *MCLM*, a to inkrementálně, takže pohyb manipulátoru je vždy o stejnou vzdálenost v daném směru. Rutina ošetřuje v každém okamžiku, jakým směrem se manipulátor může či nemůže vydat. K tomuto určení využívá pokyny převzaté z programu labyrintu pomocí proměnné *Labyrint\_IN* jak již bylo v práci zmíněno.

Pojezd v labyrintu zajišťuje volný pohyb v labyrintu, který se řídí aktuální pozicí, která je signalizována zelenou diodou na daném poli labyrintu. To ale v případě navigace správným projitím labyrintem nejde, a tak je zavedena paměť. Ta určuje, kde se nachází právě manipulátor, protože zelené diody ukazují, kam manipulátor může jet. Tím pádem program obsahuje dvě matice, které udávají aktuální a minulou navigační pozici. Tím je zajištěno kam se může manipulátor při ručním ovládání pohybovat.

Tato rutina zajišťuje i automatický pohyb, který lze zapnout taktéž ve vizualizaci. Tedy pokud je zapnuto navádění průchodu labyrintem, manipulátor automaticky projede labyrint dle jeho navigování pomocí zelených diod. Pro tuto funkci je opět využito dvou matic s pozicí a navigací, které jsou porovnávány a podle toho je zvolen směr kam se bude manipulátor pohybovat. Automatický režim jde kdykoliv během provádění vypnout a lze opět jezdit pomocí ručního ovládání.

*Paměť*  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$  Takto vypadá matice, pokud je manipulátor mimo labyrint.

*Aktual*  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$  Po zapnutí navigace se ukáže log. 1, tedy program zvolí instrukci

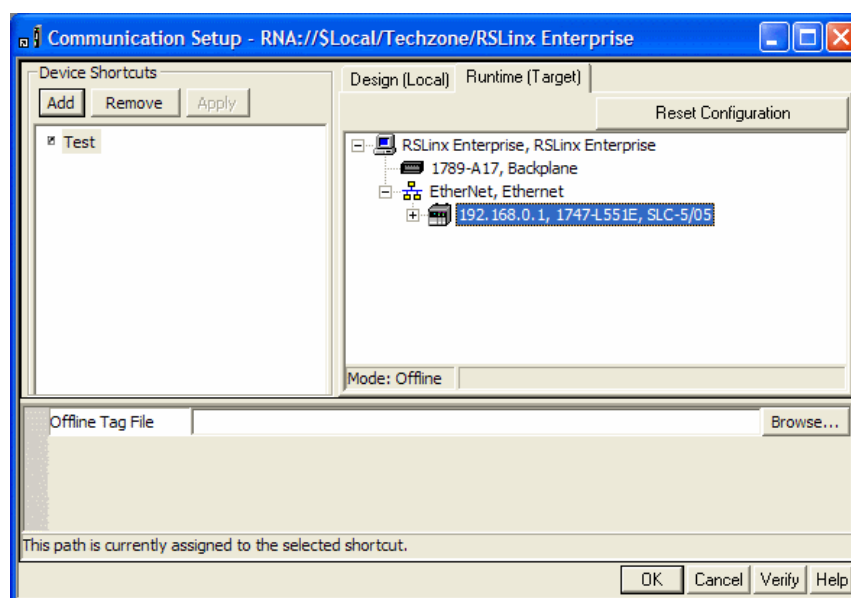
pro vjetí do labyrintu. Jakmile se zapne posuv, je přepsána paměť aktuální maticí, jelikož se hodnota ještě nezměnila. A po ukončení instrukce je hodnota matice Paměť a Aktual zase rozdílná a dojde k dalšímu vyhodnocení. Tímto způsobem je rozhodováno postupně po celou dobu zapnutí automatického režimu a navigace labyrintem. V okamžiku kdy labyrint dovoluje rozhodnout, zda půjde cestou doleva nebo doprava, je volba ponechána stále na obsluze. Pokud by byl požadavek na absolutní automatický režim, dalo by se to vyřešit generováním náhodného čísla, které by rozhodovalo o vybrání cesty.

## 4.4 VIZUALIZACE

Pro vizualizaci má firma Allen-Bradley svůj nástroj v podobě softwaru FactoryTalk View Studio, který taktéž spolupracuje s komunikací RS Linx, tudíž není těžké nakonfigurovat s kterým programovatelným automatem má spolupracovat.

### 4.4.1 Konfigurace

Nejdůležitější a vlastně jedinou povinnou částí konfigurace je zajištění komunikace mezi procesorem automatu a vizualizací, tímto krokem jsou ve vizualizaci zpřístupněny všechny proměnné procesoru a lze je využít ve vizualizaci. Stačí tedy nastavit komunikaci v záložce Communication Setup.



Obrázek 4.21 Communication Setup FactoryTalk View Studio

V Device Shortcuts přidáme procesor a názvem pro vizualizaci a následně jej vybereme v pravém okně poklepáním. Do položky Offline Tag File vložíme cestu k programu, který je i v programovatelném automatu. Tím zajistíme, že budeme mít přístup k proměnným i offline.nipulátoru i labyrintu, aby mohla vizualizace komunikovat s oběma programy.

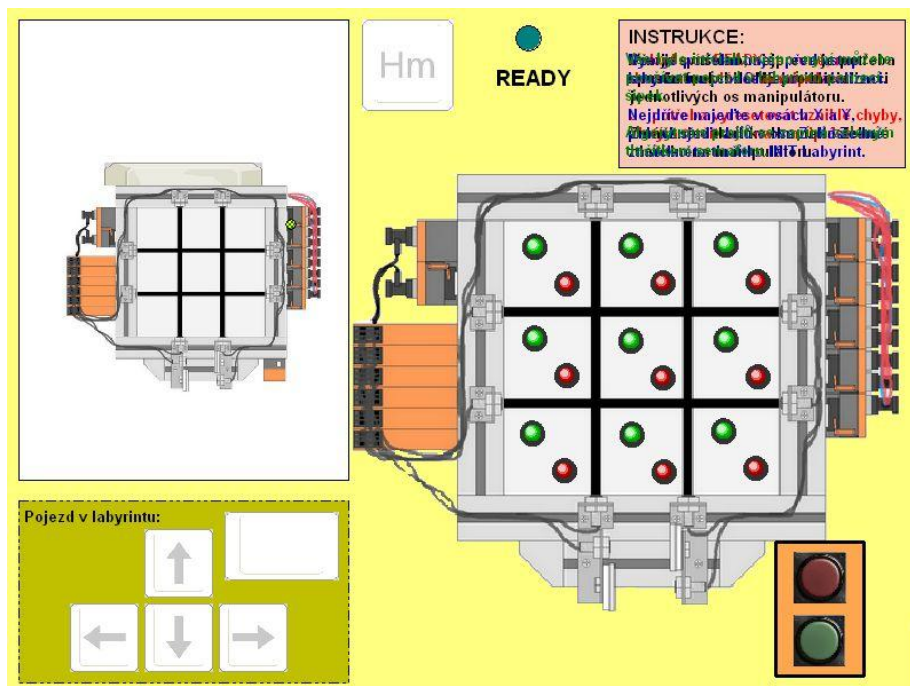
Takto přidáme procesor manipulátoru i labyrintu, aby mohla vizualizace komunikovat s oběma programy.

## 4.4.2 Vlastní vizualizace



Obrázek 4.22 Úvodní okno vizualizace

Úvodní okno pouze informuje, o jakou vizualizaci se jedná a kdo ji vytvořil, také popisuje, co vizualizace dělá.



Obrázek 4.23 Okno pro ovládání

Okno pro ovládání má několik částí. Na levé straně se zobrazuje manipulační prostor souřadného systému, a po inicializaci aktuální poloha manipulátoru a její pohyb pomocí žlutozeleného kolečka. Hned vedle se po zapnutí os zpřístupní tlačítko *Hm*, které slouží k „houmování“ os manipulátoru. O připravenosti os bez chyb informuje barevný ukazatel nad nápisem *READY*. Pokud jsou osy připraveny, svítí zeleně, pokud se vyskytne nějaká chyba, svítí červeně a zobrazí se červené hlášení v poli *INSTRUKCE*.

V poli instrukce se postupně s inicializačními kroky mění instrukce jak dále pokračovat. Nejdříve není zobrazen model labyrintu, ale místo něj ovládací kříž manipulátoru, který jsme mohli vidět již na obrázku *Obr. 4.24 Ovládací kříž pro manuální pojezd*. Ten je zpřístupněn po inicializaci manipulátoru. Pak je tedy potřeba pomocí těchto ovládacích prvků najet před vstup labyrintu a zmáčknout tlačítko *Init Labyrint*, které se zobrazí v poli aktuální polohy manipulátoru. Po úspěšném provedení této inicializace se zobrazí poloha labyrintu v souřadném systému a velký model s diodami informujícími o poloze nebo navigaci. Také se zpřístupní pole *Pojezd v labyrintu* pro ovládání pojezdu a tlačítko pro zapínání a vypínání automatického ovládání.

Po všech inicializacích je tedy v konečné vizualizaci možno vidět aktuální polohu manipulátoru v labyrintu, signalizaci navádění labyrintem pomocí diod a signalizaci vstupního semaforu. Lze ovládat pojezd manipulátoru a tlačítka vstupního semaforu labyrintu. Takže vizualizace splňuje potřebné funkce.

## 5 ZÁVĚR

Model labyrintu mi byl dodán již hotový jako práce studenta z předchozího akademického roku. Tudíž na samotné konstrukci již nebylo potřeba nic měnit. Jen bylo nutné připojit přívodní kabel, zajistit příslušné napájecí moduly a dovybavit moduly slave, které chyběly. Mou prací bylo tedy seznámit se se sběrnici AS-Interface a zprovoznit komunikaci modelu, modulu masteru AS-I, PLC a PC.

Na modelu manipulátoru pracovalo během semestru více studentů, přičemž jeden pracoval na novém zabezpečení. Já jsem model používal tak jak byl i s předešlou konfigurací v programu. Tu jsem musel lehce upravit a přizpůsobit novému zabezpečení použitému na modelu manipulátoru.

Po překonání problémů s komunikací modulu AS-I a PLC se jeví tato sběrnice jako velmi praktická a použitelná pro mnoho aplikací. Zapojení je jednoduché, jednoznačné a hlavně rychlé, díky dvoužilovému profilovanému kabelu, který zajišťuje jak komunikaci, tak napájení.

Programové prostředí RS Logix 5000 pro automaty Allen-Bradley firmy Rockwell Automation, v němž byl vytvořen program pro řízení labyrintu i manipulátoru, se mi zdá přehledné a intuitivní zároveň s jednoduchostí programování jazykem Ladder Diagram.

Samotný program labyrintu, který jsem vytvářel, funguje bez problému a vše, co měl dle zadání splňovat, splňuje. Program používá docela hodně časovačů, jistě by bylo vhodné jejich počet redukovat. S ohledem na čas, který jsem na projekt měl, však nebylo již možné program redukovat. Jistě by si to zasloužil pro větší přehlednost a jednoduchost.

Program manipulátoru mne naučil jak provazovat jednotlivé programy a jejich proměnné. Také jsem se naučil užití Motion instrukcí a podstatu řešení motorového řízení firmy Allen-Bradley. Požadavky na program jsem splnil v míře zadání, ale vím, že by se dal program stále zlepšovat. Je na něm potřeba odladit chybové stavy, tak aby nemohlo docházet ke kolizím. Vzhledem k náročnosti projektu jsem nebyl schopen všechny tyto chyby odladit. Program je velmi rozsáhlý, je to způsobeno volbou programovacího jazyka Ladder diagram. Programování v tomto jazyce bylo jednodušší, ale pro větší přehlednost, a snad i kratší kód, by bylo vhodné uvažovat o jiném druhu programovacího jazyka, například funkční diagram.

Vyzkoušel jsem si i práci s vizualizací v programu FactoryTalk View Studio. Práce s tímto programem je jednoduchá, ale některá nastavení a složitost voleb se mi zdá pro vytváření větších vizualizací nepraktické. Bohužel k programu není mnoho dostupné literatury, takže jsem nebyl schopen objevit všechny jeho možnosti. Práce s ním je v celku intuitivní.

## 6 LITERATURA

- [1] ZEŽULKA, F. A KOLEKTIV. *Programovatelné automaty* [el. text]. Brno 2003.  
Dostupné na internetu:  
[https://www.feec.vutbr.cz/et/skripta/uamt/Programovatelne\\_automaty\\_S.pdf](https://www.feec.vutbr.cz/et/skripta/uamt/Programovatelne_automaty_S.pdf)
- [2] Základy o AS-I (AS-inteface Česká republika), Zaklady\_ASI.pdf [on-line],  
[cit. 2011-05-30].  
Dostupné na [www.as-interface.cz/Seminar/Zaklady\\_ASI.pdf](http://www.as-interface.cz/Seminar/Zaklady_ASI.pdf)
- [3] AS-Interface (AS-inteface Česká republika), Interface\_V3\_0.pdf [on-line],  
[cit. 2011-05-30].  
Dostupné na [www.as-interface.cz/Seminar/AS-Interface\\_V3\\_0.pdf](http://www.as-interface.cz/Seminar/AS-Interface_V3_0.pdf)
- [4] Programovatelné logické automaty [online], [cit. 2011-05-25].  
Dostupné z <http://uprt.vscht.cz/kminekm/mrt/F5/F5k53-PLC.htm>
- [5] Becker R. AS-Interface, Řešení pro automatizaci. Vydáno AS-International Association. Překlad AS-interface Česká republika
- [6] ČSN 33 2000-4-41:2000 (EN 60204, IEC 60364-4-41), Elektronické předpisy – Elektrická zařízení – Bezpečnost: Ochrana před úrazem elektrickým proudem.
- [7] *IFM Electronic* [online]. 2009 [cit. 2011-05-25]. AC4000.  
Dostupné z <http://www.ifm.com/ifmcz/web/dsfs!AC4000.html>
- [8] *IFM Electronic* [online]. 2009 [cit. 2011-05-25]. AC5213.  
Dostupné z <http://www.ifm.com/ifmcz/web/dsfs!AC5213.html>
- [9] *IFM Electronic* [online]. 2009 [cit. 2011-05-25]. AC5210.  
Dostupné z <http://www.ifm.com/ifmcz/web/dsfs!AC5210.html>
- [10] *IFM Electronic* [online]. 2009 [cit. 2011-05-25]. AC2088.  
Dostupné z <http://www.ifm.com/ifmcz/web/dsfs!AC2088.html>
- [11] *IFM Electronic* [online]. 2009 [cit. 2011-05-25]. AC5000.  
Dostupné z <http://www.ifm.com/ifmcz/web/dsfs!AC5000.html>



- [12] *IFM Electronic* [online]. 2009 [cit. 2011-05-25]. OE0004.  
Dostupné z <http://www.ifm.com/ifmcz/web/dsfs!OE0004.html>
- [13] *IFM Electronic* [online]. 2009 [cit. 2011-05-25]. OE0003.  
Dostupné z <http://www.ifm.com/ifmcz/web/dsfs!OE0003.html>
- [14] *IFM Electronic* [online]. 2009 [cit. 2011-05-25]. OOV5012  
Dostupné z <http://www.ifm.com/ifmcz/web/dsfs!OV5012.html>
- [15] Průmyslová komunikační síť AS-interface. Automa [online]. 2004, 04,  
[cit. 2011-05-25].  
Dostupný z [http://www.odbornecasopisy.cz/index.php?id\\_document=32276](http://www.odbornecasopisy.cz/index.php?id_document=32276)
- [16] Rábová, Z., Hanáček, P., Peringer, P., Příkryl, P., Křena, B.: *Užitečné rady pro psaní odborného textu* [online].  
Brno: c1993-2008, aktualizováno 2008-11-01 [cit. 2008-11-28].  
Dostupné na URL: [http://www.fit.vutbr.cz/info/statnice/psani\\_textu.html](http://www.fit.vutbr.cz/info/statnice/psani_textu.html)
- [17] *Kinetix 2000 – User manual*. Rockwell Automation-literature. [cit. 2013-05-22].  
Dostupný z  
[http://literature.rockwellautomation.com/idc/groups/literature/documents/um/2093-um001\\_-en-p.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/um/2093-um001_-en-p.pdf)
- [18] *GuardLogix Controllers – User manual*. Rockwell Automation-literature. 2008 [cit. 2013-05-22]. Dostupný z  
[http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1756-um020\\_-en-p.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1756-um020_-en-p.pdf)
- [19] *ControlLogix Chassis – Installation instruction*. Rockwell Automation-literature. 2012 [cit. 2013-05-22]. Dostupný z  
[http://literature.rockwellautomation.com/idc/groups/literature/documents/in/1756-in005\\_-en-p.pdf](http://literature.rockwellautomation.com/idc/groups/literature/documents/in/1756-in005_-en-p.pdf)
- [20] PLC Hardware [online]. [cit. 2013-05-22], Dostupný z  
<http://www.plchardware.com/Products/RA-1756-DNB-A-UPP.aspx>
- [21] KOVÁŘ, P. Model CNC frézky. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 76s. Vedoucí diplomové práce Ing. Radek Štohl, Ph.D.

## **7 SEZNAM PŘÍLOH**

Příloha 1. CD: Elektronická verze BP

Program labyrintu

Program manipulátoru

Vizualizace pro FactoryTalk View

Reporty z programů v pdf

Report z vizualizace v pdf i html

Příloha 2. Rozložení modelu labyrintu

Příloha 3. Katalogové listy: Základní parametry vybraných prvků systému, 6s